11 - Exception

Ex. No. : 11.1 **Date: 7/06/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Write a Python program that performs division and modulo operations on two numbers provided by the user. Handle division by zero and non-numeric inputs.

Input Format:

Two lines of input, each containing a number.

Output Format:

Print the result of division and modulo operation, or an error message if an exception occurs.

For example:

Input	Result
10 2	Division result: 5.0 Modulo result: 0
7 3	Division result: 2.3333333333333335 Modulo result: 1
8 0	Error: Cannot divide or modulo by zero.

```
try:
    num1 = float(input())
    num2 = float(input())

    if num2 == 0:
        print("Error: Cannot divide or modulo by zero.")
    else:
        division_result = num1 / num2
```

```
        modulo_result = int(num1 % num2)  # Convert modulo result to integer
        print(f"Division result: {division_result}")
        print(f"Modulo result: {modulo_result}")

except ValueError:
    print("Error: Non-numeric input provided.")
```

	Input	Expected	Got	
✔	10 2	Division result: 5.0 Modulo result: 0	Division result: 5.0 Modulo result: 0	✔
✔	7 3	Division result: 2.3333333333333335 Modulo result: 1	Division result: 2.3333333333333335 Modulo result: 1	✔
✔	8 0	Error: Cannot divide or modulo by zero.	Error: Cannot divide or modulo by zero.	✔
✔	abc 5	Error: Non-numeric input provided.	Error: Non-numeric input provided.	✔

Passed all tests! ✔

Ex. No. : 11.2 **Date: 7/06/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Problem Description:

Develop a Python program that safely calculates the square root of a number provided by the user. Handle exceptions for negative inputs and non-numeric inputs.

Input Format:

User inputs a number.

Output Format:

Print the square root of the number or an error message if an exception occurs.

For example:

Input	Result
16	The square root of 16.0 is 4.00
-4	Error: Cannot calculate the square root of a negative number.
rec	Error: could not convert string to float

```
import math


try:
    num = float(input())


    if num < 0:
        print("Error: Cannot calculate the square root of a negative number.")
    else:
        sqrt_result = math.sqrt(num)
```

```
        print(f"The square root of {num} is {sqrt_result:.2f}")


except ValueError:
    print("Error: could not convert string to float")
```

	Input	Expected
✔	16	The square root of 16.0 is 4.00
✔	0	The square root of 0.0 is 0.00
✔	-4	Error: Cannot calculate the square root of a negative number.

Ex. No. : 11.3 **Date: 7/06/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Develop a Python program that safely performs division between two numbers provided by the user. Handle exceptions like division by zero and non-numeric inputs.

Input Format: Two lines of input, each containing a number.

Output Format: Print the result of the division or an error message if an exception occurs.

For example:

Input	Result
10 2	5.0
10 0	Error: Cannot divide or modulo by zero.
ten 5	Error: Non-numeric input provided.

```
def safe_division():
    try:
        # Input
        num1 = float(input())
        num2 = float(input())

        # Division
        result = num1 / num2
```

```
        # Output
        print(result)

    except ZeroDivisionError:
        print("Error: Cannot divide or modulo by zero.")
    except ValueError:
        print("Error: Non-numeric input provided.")

if __name__ == "__main__":
    safe_division()
```

	Input	Expected	Got	
✔	10 2	5.0	5.0	✔
✔	10 0	Error: Cannot divide or modulo by zero.	Error: Cannot divide or modulo by zero.	✔
✔	ten 5	Error: Non-numeric input provided.	Error: Non-numeric input provided.	✔

Passed all tests! ✔

Ex. No. : 11.4 **Date: 7/06/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Problem Description:

Write a Python script that asks the user to enter a number within a specified range (e.g., 1 to 100). Handle exceptions for invalid inputs and out-of-range numbers.

Input Format:

User inputs a number.

Output Format:

Confirm the input or print an error message if it's invalid or out of range.

For example:

Input	Result
1	Valid input.
101	Error: Number out of allowed range
rec	Error: invalid literal for int()

Program :

```
def get_number_within_range(min_val, max_val, user_input):
    try:
        number = int(user_input)
        if min_val <= number <= max_val:
            print("Valid input.")
        else:
            print("Error: Number out of allowed range")
    except ValueError:
        print("Error: invalid literal for int()")
```

```
# Define the range

min_range = 1

max_range = 100


# Get user input

user_input = input().format(min_range, max_range)


# Call the function to check if the input is within the specified range

get_number_within_range(min_range, max_range, user_input)
```

	Input	Expected	Got	
✔	1	Valid input.	Valid input.	✔
✔	100	Valid input.	Valid input.	✔
✔	101	Error: Number out of allowed range	Error: Number out of allowed range	✔

Passed all tests! ✔

Ex. No. : 11.5 **Date: 7/06/2024**

Register No.: 231401001 **Name: Aafrin Fathima N**

Problem Description:

Write a Python program that asks the user for their age and prints a message based on the age. Ensure that the program handles cases where the input is not a valid integer.

Input Format:

A single line input representing the user's age.

Output Format:

Print a message based on the age or an error if the input is invalid.

For example:

Input	Result
25	You are 25 years old.
rec	Error: Please enter a valid age.
-5	Error: Please enter a valid age.

```
def print_age_message(age):
    if age is None:
        print("Error: Please enter a valid age.")
    elif age < 0:
        print("Error: Please enter a valid age.")
    else:
        print("You are {} years old.".format(age))
```

```
try:
    user_input = input()
    user_age = int(user_input)
    print_age_message(user_age)
except ValueError:
    print("Error: Please enter a valid age.")
except EOFError:
    print("Error: Please enter a valid age.")
```

	Input	Expected	Got	
✔	25	You are 25 years old.	You are 25 years old.	✔
✔	rec	Error: Please enter a valid age.	Error: Please enter a valid age.	✔
✔	!@#	Error: Please enter a valid age.	Error: Please enter a valid age.	✔

Passed all tests! ✔